UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/468,051	12/20/1999	THOMAS D. HARTNETT	RA-5271	3159

7590 12/31/2002

UNISYS CORPROATION
ATTN BETH L MCMAHON
M S 4773
P O BOX 64942
ST. PAUL, MN 551640942

EXAMINER	
WOOD, WILLIAM H	
ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 12/31/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/468,051	HARTNETT ET AL.	
	Examiner	Art Unit	
	William H. Wood	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 October 2002* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Claims 1-20 have been examined.

Claim Rejections - 35 USC § 112

1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. Claims 1-20 are rejected under 35 U.S.C. 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Applicant amended into each of the independent claims 1, 8 and 14 the limitation fetch circuit coupled *directly* to the execution circuit. This limitation was not found in the prior disclosure of the invention.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1, 8-12, 14 and 16-20 are rejected under 35 U.S.C. 102(b) as being anticipated by McLellan (USPN 5,325,495).

In regard to claim 1, McLellan taught the following limitations:

i) *For use in an instruction processor that executes instructions included in a predetermined instruction set* (column 1, lines 10-15; computer systems include instruction processors with a predetermined instruction set)

ii) *instruction processor executes at an execution rate determined by a system clock signal* (Figure 2, shows instructions advancing based on system clock cycles)

iii) *a pipeline execution circuit* (Figure 1, pipeline stage 4 and beyond; column 5, lines 22-23) *to process a first predetermined number of instructions simultaneously* (the number of instructions to be processed is the number of stages the pipeline may have; pipelines by nature are executing various instructions simultaneously), *each of the first predetermined number of instructions being in a respectively different stage of execution within the pipeline execution circuit* (Figure 2)

iv) *instructions being capable of advancing to a next stage of execution within the pipeline execution circuit at a time determined by the clock signal* (Figure 2, shows instructions advancing based on system clock cycles)

v) *a pipeline fetch circuit coupled directly to the pipeline execution circuit to process a second predetermined number of instructions simultaneously* (Figure 1, pipeline stages 1-3 and part 14; the first sections of this pipeline are the fetch circuit), *each of the second predetermined number of instructions being in a respectively different stage of processing within the pipeline fetch circuit* (Figure 2), *an instruction being capable of advancing to a next stage of execution*

independently of the times at which instructions advance to a next stage of execution within the pipeline execution circuit (column 4, lines 32-36; by avoiding stalling the fetch stages, they are now independent of the execution stages which have stalled and thus instructions advance)

vi) *an instruction being capable of advancing to a next stage of execution within the pipeline fetch circuit at a time determined by the system clock signal* (Figure 2, shows instructions advancing based on system clock cycles)

In regard to claim 8, McLellan taught the following limitations

i) *For use in an instruction processor that executes instructions of a machine instruction set* (column 1, lines 10-15; computer systems include instruction processors with a predetermined instruction set)

ii) *a synchronous pipeline system* (Figure 2, shows instructions advancing based on system clock cycles; or synchronous pipeline)

iii) *a plurality of execution logic sections* (Figure 1, pipeline stage 4 and beyond; column 5, lines 22-23) *each of the execution logic sections being coupled to at least one respective other one of the execution logic sections* (Figure 1), *each of the execution logic sections to perform a predetermined stage of execution of any of the instructions* (column 5, lines 3-6; indicates the sections have a predetermined stage value)

iv) *whereby each of the execution logic sections is capable of receiving a new instruction to process at predetermined time increments* (Figure 2, shows instructions advancing based on system clock cycles (predetermined time))

v) *a plurality of fetch logic sections* (Figure 1, pipeline stages 1-3 and part 14; the first sections of this pipeline are the fetch circuit) *wherein at least one of the plurality of fetch logic sections is coupled directly to at least one of the plurality of execution logic sections* (Figure 1, coupled through the q-stage), *each of the fetch logic sections being coupled to at least one respective other one of the fetch logic sections* (Figure 1), *each of the fetch logic sections to perform a predetermined pre-execution stage of instruction execution* (column 5, lines 3-6; indicates the sections have a predetermined stage value; also the decoder is actually labeled as a decoder function), *each of the fetch logic sections being capable of receiving a new instruction to process in a manner that is independent of whether any of the plurality of execution logic sections receives a new instruction to process* (column 4, lines 32-36; by avoiding stalling the fetch stages, they are now independent of the execution stages which have stalled)

vi) *each of the fetch logic sections being capable of receiving a new instruction to process at the predetermined time increments* (Figure 2, shows instructions advancing based on system clock cycles (predetermined time))

In regard to claim 9, McLellan taught the limitation *passing at predetermined time increment* (Figure 2, shows instruction passing based on system clock cycles). McLellan did not explicitly state the limitations:

i) *further including a select circuit coupled to one of the plurality of fetch logic sections to allow any instruction to be passed between first and second ones of the plurality of fetch logic sections*

ii) *passing if the second one of the plurality of fetch logic sections is not executing an instruction prior to the predetermined time increment*

Clearly, the first stage has some mechanism (selection circuit) to allow (or select) an instruction to be received into the stage to be operated on if the following stage (second stage) is ready to receive the instructions. This is the nature of pipeline systems, instructions continually progress down the line from stage to stage. If the second logic circuit had itself stalled, the pipeline would not allow the instruction to progress. Therefore, a circuit to allow for the selection of passing or not passing between to circuits (stages) in a pipeline would have been inherent to the pipeline system formed by McLellan and Sites.

In regard to claim 10, McLellan taught the limitation *passing at the predetermined time* increment (Figure 2, shows instruction passing based on system clock cycles). McLellan did not explicitly state the limitations:

i) *wherein the select circuit includes a control circuit to further allow any instruction to be passed between the first and second ones of the plurality of fetch logic section*

ii) *passing if the second one of the plurality of fetch logic section is executing an instruction while a third predetermined one of the plurality of fetch logic sections is not executing an instruction*

Clearly, the first stage has some mechanism (selection circuit) to allow an instruction to be received into the stage to be operated on if the following stage (second stage) is ready to receive the instructions by allowing the instruction to move on down the pipeline to the available third stage. This is the nature of pipeline systems, instructions continually progress down the line from stage to stage. If the third logic section is ready to receive the second logic section's instruction then the pipeline can continue to operate smoothly. Therefore, a circuit to allow for the selection of passing or not passing between to circuits (stages) in a pipeline would have been inherent to the pipeline system formed by McLellan and Sites.

In regard to claim 11, McLellan taught the following:

i) *passing if prior to the predetermined time increment a predetermined one of the plurality of execution logic sections is performing a predetermined function* (column 4, lines 32-36; by avoiding stalling the stages, they are now independent of the execution stages are performing a predetermined function by being stalled)

ii) *passing at the predetermined time increment* (Figure 2, shows instruction passing based on system clock cycles)

McLellan did not explicitly state:

iii) *wherein the select circuit includes a control circuit to allow any instruction to be passed between the first and second ones of the plurality of fetch logic sections*

Clearly, the first stage has some mechanism (selection circuit) to allow (or select) an instruction to be received into the first stage to be operated on if the first stage is not busy (stalled). This is the nature of pipeline systems, instructions continually progress down the line from stage to stage. If the first logic circuit had itself stalled, the pipeline would not allow the instruction to progress. Therefore, a circuit to allow for the selection of passing or not passing between to circuits (stages) in a pipeline would have been inherent to the pipeline system formed by McLellan and Sites.

In regard to claim 12, McLellan taught the limitation *passing at the predetermined time increment* (Figure 2, shows instruction passing based on system clock cycles).

McLellan did not explicitly state:

i) *further including a second select circuit coupled to the second one of the plurality of fetch logic sections to allow any instruction to be passed between the second one and a third one of the plurality of fetch logic sections*

ii) *passing if the third one of the plurality of fetch logic sections is not executing an instruction prior to the predetermined time increment*

Clearly, the third stage (circuit) has some mechanism (selection circuit) to allow (or select) an instruction to be received into the stage to be operated on if the third stage (circuit) is ready to receive the instructions (does not already have an instruction). This is the nature of pipeline systems, instructions continually progress down the line from stage to stage. If the third logic circuit had itself stalled, the pipeline would not allow the instruction to progress. Therefore, a circuit to allow for the selection of passing or not passing between to circuits (stages) in a pipeline would have been inherent to the pipeline system formed by McLellan and Sites.

In regard to claim 14, McLellan taught the limitations:

i) *for use in an instruction processor having a synchronous instruction pipeline that executes instructions at a rate determined by a system clock* (Figure 2, shows instructions advancing based on system clock cycles; or synchronous pipeline)

ii) *the instruction pipeline including a predetermined number of execution logic sections coupled to each other in sequence* (Figure 1, pipeline stage 4 and beyond; column 5, lines 22-23), *each to perform a respectively different stage of execution on any instruction* (Figure 2), *and a predetermined number of fetch logic sections coupled to each other in sequence* (Figure 1, pipeline stages 1-3 and part 14; the first sections of this pipeline are the fetch circuit), *each to perform a respectively different stage of pre-execution on any instruction* (column 5, lines 3-6; indicates the sections have a predetermined stage value; also the

decoder is actually labeled as a decoder function), *and wherein at least one of the fetch logic sections is coupled to at least one of the execution logic sections* (Figure 1, coupled through the q-stage)

iii) *(a) providing at least one of the fetch logic sections that is coupled directly to at least one of the execution logic sections* (Figure 1)

iv) *(b) processing a respective one of the instructions by each of the execution logic sections for a first predetermined time period* (definition of a pipeline and clearly present in McLellan's pipeline)

v) *(c) allowing ones of the execution logic sections to each pass the respective one of the instruction to another coupled one of the execution logic sections after the first predetermined time period elapses* (definition of a pipeline and clearly present in McLellan's pipeline)

vi) *(d) allowing at least one of the execution logic sections to retain the respective instruction for longer than the first predetermined time period* (the definition of a pipeline stall and clearly present in McLellan; see column 4, lines 32-38)

vii) *(e) allowing one of the fetch logic sections each to begin processing a respective instruction during a subsequent predetermined time period that is subsequent to the first predetermined time period if each of the fetch logic sections was not processing a respective instruction during the first predetermined time period* (column 4, lines 32-36; by avoiding stalling the stages, they are now independent of the execution stages which have stalled and thus the fetch stages are capable of processing instructions in following

predetermined time periods provided the stages are not stalled themselves)

In regard to claim 16, McLellan did not explicitly state the limitations:

i) *wherein one of the execution logic sections includes logic to retrieve instruction operands required to execute an instruction*

ii) *wherein step (c) includes the step of allowing the one of the execution logic sections to retain the respective instruction so that a time period that is longer than the predetermined time period may be utilized to retrieve an operand required for execution of the respective instruction*

Despite McLellan not specifically stating the limitations above, McLellan is clearly aware of the idea of missing operands causing a stall in a pipeline, which is what this claim is interpreted as. McLellan's analogy of a car in an assembly line gives evidence of this in column 3, lines 5-7. Therefore, the concept of stalling while fetching operands is inherent in McLellan.

In regard to claim 17, McLellan taught the following:

i) *wherein step (d) includes the step of allowing a predetermined one of the fetch logic section to begin decode processing of a respective instruction after the predetermined time period elapses if each of the fetch logic sections was not processing a respective instruction during the predetermined time period*

This limitation simply states the decode stage is operable during the stalled execution stages of the pipeline. McLellan allowed for this in column 4, lines 32-36; by avoiding

stalling the fetch stages, they are now independent of the execution stages which have stalled and thus the fetch stages are capable of processing instructions. Clearly the stages have to not be processing instructions if the new instructions are to advance.

In regard to claim 18, McLellan further taught the limitation *further including the step of (e) allowing a predetermined one of the fetch logic sections to retrieve additional instructions to prepare for executing the fetch instructions.* This limitation simply states the fetch logic can continue to retrieve instructions to process. McLellan allowed for this in column 4, lines 32-36; by avoiding stalling the stages, they are now independent of the execution stages which have stalled and thus the fetch stages are capable of retrieving instructions to process.

In regard to claim 19, McLellan taught:

> i) *instruction retrieval irrespective of whether any other of the fetch logic sections or the execution logic sections begins processing another instruction during any subsequent predetermined time period that is subsequent to the first predetermined time period* (column 4, lines 32-36; by avoiding stalling the stages, they are now independent of the execution stages which have stalled and thus the fetch stages are capable of retrieving instructions to process)

McLellan did not explicitly state:

> ii) *wherein step (e) is repeated until a predetermined maximum number of instructions is retrieved*

However, McLellan's invention uses a queue in the Q-stage. This clearly has a maximum value, which is predetermined (column 6, lines 39-48; McLellan's invention defaults to a value of 1). Clearly, McLellan's invention will repeat until the queue is filled as long as there is a stall in the pipeline execution stages. Therefore, repeating until a maximum number of instructions are retrieved is inherent in the design of McLellan's invention.

In regard to claim 20, McLellan did not explicitly state:

i) *further including the step of (e) allowing ones of the fetch logic sections to begin processing a respective instruction during an additional subsequent predetermined time period if each of the fetch logic sections was not processing a respective instruction during the most recently elapsed predetermined time period*

ii) *further including the step of (f) repeating step (e) until each of the fetch logic sections is processing a respective instruction*

Clearly, in order to keep the entire pipeline from stalling the fetch portion of the pipeline in McLellan needs to continue to begin processing instructions. Refer to column 4, lines 32-36 where McLellan reveals the desire to keep the entire pipeline from stalling. To this end, it is inherent that McLellan continues to fetch and begin processing instructions until it can no longer do so (the queue is full) and all the fetch logic sections are processing an instruction.

Cl im Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 2-6 are rejected under 35 U.S.C. 103(a) as being unpatentable over McLellan (USPN 5,325,495) in view of Sites et al. (USPN 5,778,423).

In regard to claim 2, McLellan did not explicitly state the limitation *wherein the pipeline fetch circuit includes an instruction queue to store a predetermined maximum number of the instructions that are ready to be processed by the pipeline fetch circuit.* However, a data structure (the queue is interpreted as this) to hold information waiting to be processed by a circuit or stage in a pipeline is a well known concept to those of ordinary skill in the art. Furthermore, Sites taught a data structure to hold a maximum number of instructions waiting for processing (Figure 1, part 21). McLellan offers the motivation to include this pipeline configuration of Sites by indicating the McLellan computer system be the type described in the Sites patent (column 4, line 67 to column 5, line 3; column 5, lines 22-25). Therefore, it would have been obvious to one of ordinary skill in the art to implement a data structure for waiting instructions before they are processed.

In regard to claim 3, McLellan taught the limitation *wherein an instruction can enter a stage of processing independently of the movement of instructions through the pipeline execution circuit* (column 4, lines 32-36; by avoiding stalling the stages, they are now

independent of the execution stages which have stalled). McLellan did not explicitly state *wherein the pipeline fetch circuit includes a pre-decode logic circuit to generate pre-decode signals for an instruction that is in a pre-decode stage of processing within the pipeline fetch circuit*. However, pre-decode stages are common to pipelines. Practitioners of the art would recognize numerous combinations of fetch, pre-decode, and decode stages in pipelines as standard procedure. Furthermore, Sites taught a pre-decode stage (column 10, lines 12-21; S1 in this case is the pre-decode stage). McLellan offers the motivation to include this pipeline configuration of Sites by indicating the McLellan computer system be the type described in the Sites patent (column 4, line 67 to column 5, line 3; column 5, lines 22-25). Therefore, it would have been obvious to one of ordinary skill in the art to implement a pre-decode logic circuit (stage) to generate pre-decode signals in McLellan.

In regard to claim 4, McLellan taught the limitations:

i) *wherein the pipeline fetch circuit includes a decode logic circuit* (Figure 1, part 14)

ii) *decode circuit to generate decode signals for an instruction that is in a decode stage of processing within the pipeline fetch circuit* (column 5, lines 7-11; decoder is generating decode signals)

iii) *wherein an instruction can pass between the two stages of processing independently of the movement of instructions through the pipeline execution*

circuit (column 4, lines 32-36; by avoiding stalling the stages, they are now independent of the execution stages which have stalled)

Though McLellan does not explicitly show a decode stage coupled as described by the limitation *decode logic circuit coupled to the pre-decode logic circuit*, practitioners in the art would recognize many possible configurations of pipeline stages in pipeline circuits. Furthermore, Sites taught decode circuit coupled to the pre-decode circuit (column 10, lines 12-21; S2 is the decode circuit which is coupled to the pre-decode circuit, S1). McLellan offers the motivation to include this pipeline configuration of Sites by indicating the McLellan computer system be the type described in the Sites patent (column 4, line 67 to column 5, line 3; column 5, lines 22-25). Therefore, it would have been obvious to one of ordinary skill in the art to implement a decode stage coupled to a pre-decode stage as described in Sites.

In regard to claim 5, McLellan taught *instruction received at a time determined by the system clock signal* (Figure 2, shows instructions advancing based on system clock cycles). McLellan did not explicitly state the following two limitations:

i) *wherein the pipeline fetch circuit includes a first selection circuit coupled to the pre-decode logic circuit to allow an instruction to be received by the pre-decode logic circuit*

ii) *instruction received if the decode logic circuit is available to accept an instruction currently being executed by the pre-decode logic circuit*

Given the above combination of Sites, clearly the pre-decode stage is coupled to some mechanism (selection circuit) to allow (or select) an instruction to be received into the stage to be operated on if the following decode stage is ready to receive the instruction the pre-decode stage is currently operating on. This is the nature of pipeline systems, instructions continually progress down the line from stage to stage. If the decode logic circuit had itself stalled, the pipeline would not allow the instruction to progress. Therefore, a selection circuit for allowing an instruction to be received by the pre-decode circuit if the decode circuit is available to accept an instruction would have been inherent to the pipeline system formed by McLellan and Sites.

In regard to claim 6, McLellan taught the *allowing an instruction to enter the decode stage of execution at a time determined by the system clock signal* (Figure 2, shows instructions advancing based on system clock cycles). McLellan did not explicitly state the following limitations:

i) *wherein the pipeline fetch circuit includes a second selection circuit coupled to the decode logic circuit*

ii) *second selection circuit to allow an instruction to enter the decode stage of execution if the decode logic circuit is not processing another instruction*

Given the above combination of Sites, clearly the decode stage is coupled to some mechanism (selection circuit) to allow (or select) an instruction to be received into the decode stage to be operated on if the decode stage is ready to receive the instruction by not currently processing an instruction. This is the nature of pipeline systems,

instructions continually progress down the line from stage to stage. If the decode logic circuit had stalled, the pipeline would not allow the instruction to progress. Therefore, a selection circuit for allowing an instruction to be received by the decode circuit if the decode circuit is available to accept an instruction would have been inherent to the pipeline system formed by McLellan and Sites.

7. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over McLellan (USPN 5,325,495) in view of Sites et al. (USPN 5,778,423) as applied to claim 5 and in further view of Alferness et al. (USPN 5,577,259).

In regard to claim 7, McLellan did not teach the following:

i) *wherein the pipeline execution circuit includes a microcode-controlled sequencer*

ii) *sequencer to control execution of the extended stages of execution of extended-mode ones of instructions*

iii) *wherein during the extended stages of execution, ones of the instructions being executed by the pipeline execution circuit are not advancing to a next stage of execution within the pipeline execution circuit*

iv) *wherein the first selection circuit includes a control circuit to allow an instruction to enter the pre-decode stage of processing while the extended-mode ones of the instructions are not advancing to a next stage of execution within the pipeline execution circuit*

These limitations can be found in Alferness, who taught *a microcode-controlled sequencer* (column 1, lines 24-31) and the *sequencer to control extended stages of execution of extended-mode instructions* (column 1, lines 26-31; extended cycle instructions are the extended-mode instructions). Furthermore, Alferness taught limitation iii) in column 4, lines 63-67. When the microcode system of Alferness, which is combined in the execution stages of McLellan, processes extended cycle instructions and halts (as it necessarily does during extended cycle instruction execution), the fetch stages of the combined inventions (including pre-decode) would continue to process instructions, since these stages operate independent of the execution stages (as already shown). Alferness also demonstrated why having such a microcode-controlled sequencer is desirable (column 2, lines 29-42). Therefore, it would have been obvious to one of ordinary skill in the art to include a microcode system as found in Alferness in McLellan in order to design a much more flexible and easy to operate processing system.

8. Claims 13 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over McLellan (USPN 5,325,495) in view of Alferness et al. (USPN 5,577,259).

In regard to claim 13, McLellan did not teach the following two limitations:

i) *further including a microcode controlled logic section coupled to at least one of the execution logic sections to insert additional extended stages of instruction*

execution for each of the predetermined ones of the instructions that are extended-mode instructions

ii) *whereby each of the fetch logic sections include circuits to allow a new instruction to be received by one or more of the fetch logic sections at the predetermined time increments during the additional extended stages of instruction execution if each of the fetch logic sections is not already processing one of the instructions*

These limitations can be found in Alferness, who taught *a microcode-controlled sequencer* (column 1, lines 24-31) and the *sequencer to insert additional extended stages of instruction execution for each of the predetermined ones of the instructions that are extended-mode instructions* (column 2, lines 29-33). Furthermore, Alferness taught limitation ii) in column 4, lines 63-67. When the microcode system of Alferness, which is combined in the execution stages of McLellan, processes extended cycle instructions and halts (as it necessarily does during extended cycle instruction execution), the fetch stages of the combined inventions (including pre-decode) would continue to process instructions, since these stages operate independent of the execution stages (as already shown). Alferness also demonstrated why having such a microcode-controlled sequencer is desirable (column 2, lines 29-42). Therefore, it would have been obvious to one of ordinary skill in the art to include a microcode system as found in Alferness in McLellan in order to design a much more flexible and easy to operate processing system.

In regard to claim 15, McLellan did not teach the following limitations

i) *wherein one of the execution logic sections is a microcode controller sequencer*

ii) *wherein step (c) includes the step of allowing the microcode controlled sequencer to retain the respective instruction for the purpose of performing additional extended-mode execution cycles for the respective instruction*

These limitations can be found in Alferness, who taught *a microcode-controlled sequencer* (column 1, lines 24-31) and the *sequencer to retain extended stages of execution of extended-mode instructions* (column 4, lines 63-67). Alferness also demonstrated why having such a microcode-controlled sequencer is desirable (column 2, lines 29-42). Therefore, it would have been obvious to one of ordinary skill in the art to include a microcode system as found in Alferness in McLellan in order to design a much more flexible and easy to operate processing system.

Remarks

9. Examiner has repeated the rejection of the previous Office Action (paper number 3) above, taking into account the additional amended limitations.

10. Examiner has considered Applicant's arguments with regard to claims 1, 8 and 14 and found them not to be persuasive. Applicant interpreted the Examiner's rejection of the previous Office Action as stating the Q-stage of McLellan was included in the pipeline fetch circuit. This was not the interpretation intended on page 4, column 6 of the previous Office Action. Applicant then correctly interpreted the Examiner's rejection as stating the Q-stage is not included in the pipeline fetch circuit. The first interpretation

does not apply to the rejection and Applicant argued with regard to the second interpretation: the fetch circuit of McLellan is not directly coupled to the execution circuit. Examiner disagrees. Upon close inspection of the Figure 1's Q-stage, one can clearly see that the last stage of the fetch circuit (stage 3) goes directly to the first stage of the execution circuit (stage 4), notice the arrow going through the multiplexor 18. The fact that instructions can be side tracked in the queue 16 does not negate the fact that the fetch and execution circuits have a direct connection as well. Therefore, the original rejection set forth in the previous Office Action is maintained.

11. Applicant's arguments with regard to claim 9 have been fully considered but they are not persuasive. The broadest reasonable interpretation of the claim language is simply to advance instructions to the next stage if that stage is available for processing. This concept is argued as inherent by examiner and proven so by Hayes (page 592, section 7.2.1) and McLellan (column 1-4; indicate stalls happen in piplines). Applicant argues to the contrary and Examiner disagrees with Applicant.

12. Applicant's arguments with regard to claim 10 have been fully considered but they are not persuasive. The broadest reasonable interpretation of the claim language is predicting a pipeline will continue to pass instructions if there is no indication of a stall. This is the definition of pipelines. See also Hayes (page 592, section 7.2.1) and McLellan (column 1-4; indicate stalls happen in piplines). Applicant argues to the contrary and Examiner disagrees with Applicant.

13. Applicant's arguments with regard to claim 11 have been fully considered but they are not persuasive. The broadest reasonable interpretation of the claim language

is simply that instructions continue moving in the fetch section if the execution section is stalled (or some other function, which is causing the stall). Examiner argues this is inherently the case with the system of McLellan since it focuses on separating the two halves of the pipeline. Furthermore, this is actually found indicated in McLellan (column 1-4; discusses pipeline stalls and how to avoid them). Applicant argues lack of inherency and Examiner maintains his argument.

14. Applicant's arguments with regard to claim 12 have been fully considered but they are not persuasive. This is the same argument as above for claim 9, only applied to two different stages. Examiner maintains the above argument.

15. Applicant's arguments with regard to claim 16 have been fully considered but they are not persuasive. Applicant argues that it is not inherent to pipeline design to allow a stage of execution to hold an instruction longer in order to retrieve an operand. Applicant further states that an example of a cache that is large enough to prevent cache misses for all application being executed, no stall is ever created. This is however not the way in which caches are designed. It is not practical to build such a large cache. Cache are designed cost effectively in such a manner that the optimum size is utilized still allowing for a certain small percentage of misses to occur. If a miss should occur a pipeline stage must be equipped to handle it and take longer to proceed than expected. This concept is inherent to pipeline design as evidenced by Hayes page 377, paragraph above 4 bulleted points and the bulleted points themselves. Therefore, Examiner maintains the original argument.

16. Applicant's arguments with regard to claim 7, 13 and 15 have been fully considered but they are not persuasive. Applicant argues Alferness teaches away from the claimed invention in that it fails to provide a control circuit that allows the fetch section to continue fetching instructions if the extended-mode instructions are not advancing. This might be true for Alferness taken alone, however in the combination of McLellan and Alferness this argument has no merit. It would have been obvious to one of ordinary skill in the art that in such a combination McLellan's ability to avoid stalling the fetch stages and continue fetching instructions would be included. This is one of the great contributions of McLellan and cannot be arbitrarily removed. Furthermore McLellan's system actually helps handle the dynamic conditions of the pipeline by decoupling the front and back ends, thus negating the statement on column 4, lines 59-67.

17. Applicant's arguments with regard to claim 19 have been fully considered but they are not persuasive. Applicant argues it is not inherent to repeatedly fetch for a predetermined number of instructions irrespective of whether the fetch and execution sections begins processing another instruction during a subsequent predetermined period of time to the first predetermined time period. Examiner maintains the previous argument that this is inherently the case by the fact the queue is being filled in the Q-stage. Furthermore, in response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., address generation section 111 and instruction queue 218) are not recited in the rejected claim(s). Although the claims are interpreted in light of the

specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Finally, Examiner indicates not only is it inherent as read from the McLellan reference, but a maximum predetermined number can still be interpreted as 1, indicating regardless of queues the McLellan reference still overcomes Applicant's claims.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to William H. Wood whose telephone number is (703)305-3305. The examiner can normally be reached 7:30am - 5:00pm Monday thru Thursday and 7:30am - 4:00pm every other Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662. The fax phone numbers for the organization where this application or proceeding is assigned are (703)746-7239 for regular communications and (703)746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.

William H. Wood
December 18, 2002

Kakali Chaki

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100